UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P O Box 1450
Alexandria, Virginia 22313-1450
www uspto gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/549,542	09/15/2005	Peter Rostin	4414-38	1651

80167 7590 09/25/2008

Ryan, Mason & Lewis, LLP
90 Forest Avenue
Locust Valley, NY 11560

EXAMINER	
HO, VIRGINIA T	

ART UNIT	PAPER NUMBER
4148	

MAIL DATE	DELIVERY MODE
09/25/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/549,542	ROSTIN ET AL.	
	Examiner	Art Unit	
	VIRGINIA HO	4148	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *15 September 2005*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-40* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-40* is/are rejected.

7)☒ Claim(s) *22* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *09/15/2005*.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

1. The instant application having Application No. 10/549,542 filed on 10/28/2005 is presented for examination by the examiner.

Specification

2. The use of the trademark RSA SECURID® has been noted in this application. It should be capitalized wherever it appears and be accompanied by the generic terminology.

Although the use of trademarks is permissible in patent applications, the proprietary nature of the marks should be respected and every effort made to prevent their use in any manner which might adversely affect their validity as trademarks.

3. The use of the trademark RSA ACE/SERVER® has been noted in this application. It should be capitalized wherever it appears and be accompanied by the generic terminology.

Although the use of trademarks is permissible in patent applications, the proprietary nature of the marks should be respected and every effort made to prevent their use in any manner which might adversely affect their validity as trademarks.

4. The use of the trademark RSA ACE/AGENT® has been noted in this application. It should be capitalized wherever it appears and **be accompanied by the generic terminology**.

Although the use of trademarks is permissible in patent applications, the proprietary nature of the marks should be respected and every effort made to prevent their use in any manner which might adversely affect their validity as trademarks.

Claim Rejections - 35 USC § 112

5. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

6. Claim 15 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claim 15 comprises of <u>the method of claim 1 wherein the seed generation server initiates the seed generation process responsive to receipt of a management command.</u> The term "***management command***" renders the claim indefinite as the term does not have a standard and well-established meaning set forth in the prior art, nor does the term have a sufficiently clear definition in the specification.

Claim Rejections - 35 USC § 103

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

8. Claims 1, 5-6, 13, 19, and 35-40 are rejected under 35 U.S.C. 103(a) as being unpatentable over Spies et al. *(US Patent No. 6230269) (hereinafter Spies).*

As per claim 1, Spies teaches the <u>method for secure generation of a seed for use in performing one or more cryptographic operations, the method comprising the steps of</u>: a CLIENT <u>providing a first string</u> to a server *(column 2, lines 58-67, the client sends a string P to*

the server within an encrypted message), the SERVER generating a second string, encrypting the second string, and sending the encrypted second string to the CLIENT *(column 6, lines 53-55, the server generates string, S and sends it to the client; column 8, lines 4-6, the string P and the string S are exchanged between the client and server in an encrypted form)*; the CLIENT decrypting the encrypted second string; and the client generating the seed as a function of at least the first string and the second string *(column 7, lines 41-44, four values, including the two strings P and S which were exchanged between the client and server, are combined to serve as the seed.)*.

Spies teaches the process of generating a seed initiated by a client. Additionally, Spies teaches the process initiated by a server, as the resulting outcome is the same regardless of whether the client or the server started the seed generation process by providing the first string.

Spies does not teach the method wherein the server independently generates the seed as a function of at least the first string and the second string.

However, it would have been obvious for one of ordinary skill in the art at the time of the invention for the server to independently generate the seed as a function of at least the first string and the second string, in order to validate that the seed was generated by the client correctly (by checking the seed generated by the client against that of the server). One would have been motivated to do so, as it would increase the security of the seed generation protocol by ensuring that neither of the strings generated by the client and servers were intercepted and replaced by an attacker. In doing so, both the client and the server could be assured that the other received the correct string.

As per claim 5, Spies teaches the method of claim 1. Spies further teaches the method wherein the key utilized by the seed generation client to encrypt the second string comprises a public key of the seed generation server *(column 2, lines 65-67, Spies teaches that the key used to encrypt the message containing the string generated by the client is the server's public key).*

As per claim 6, Spies teaches the method of claim 1. Spies further teaches the method wherein the key utilized to encrypt the FIRST string comprises a secret key shared by the seed generation client and the seed generation server *(column 3, lines 11-13, the server encrypts the string with a session key shared by the client and server).* It is clear that while Spies teaches encrypting the first string generated by the server with the secret key, it would have been equally possible to encrypt the second string generated by the client with the secret key shared by the client and server, as both have access to the same key.

As per claim 13, Spies teaches the method of claim 1. Spies further teaches the method wherein the seed generation client is associated with a first processing device and the seed generation server is associated with a second processing device *(column 4, lines 23 and 60, both the client and the server have a processor and a memory).* Therefore Spies teaches a client and a server which are each associated with processing devices as defined within the specification *(Figure 2; the configuration diagram of the client/server features a CPU and memory).*

As per claim 19, Spies teaches the method of claim 1. Spies further teaches the method wherein the FIRST string comprises a combination of at least two component strings, including at least a first component generated in the seed generation client by interaction with the seed generation server and a second component previously stored in the seed generation client

(column 6, lines 59-62, the string S generated by the server is a function of two values received from the client as well as a private value of the server).

Spies teaches the process of generating a seed initiated by a client. Additionally, Spies teaches the process initiated by a server, as the resulting outcome is the same regardless of whether the client or the server started the seed generation process by providing the first string.

As per claim 35, Spies teaches an apparatus for secure generation of a seed for use in performing one or more cryptographic operations, the apparatus comprising: a processing device comprising a process coupled to a memory, the processing device implementing at least one of a seed generation client and a seed generation server *(column 4, lines 23 and 60, both the client and the server have a processor and a memory)*; the CLIENT provides a first string to the SERVER *(column 2, lines 58-67, the client sends a string P to the server within an encrypted message)*; the SERVER generates a second string, encrypts the second string utilizing a key, and sends the encrypted second string to the CLIENT *(column 6, lines 53-55, the server generates string, S and sends it to the client; column 8, lines 4-6, the string P and the string S are exchanged between the client and server in an encrypted form)*; and the CLIENT decrypts the encrypted second string and independently generates the seed as a function of at least the first string and the second string *(column 7, lines 41-44, four values, including the two strings P and S which were exchanged between the client and server, are combined to serve as the seed).*

Spies teaches the process of generating a seed initiated by a client. Additionally, Spies teaches the method wherein the seed generation server provides a first string to the seed generation client, as the resulting outcome is the same regardless of whether the client or the server started the seed generation process by providing the first string.

Spies does not teach the method wherein the server independently generates the seed as a function of at least the first string and the second string.

However, it would have been obvious for one of ordinary skill in the art at the time of the invention for the server to independently generate the seed as a function of at least the first string and the second string, in order to validate that the seed was generated by the client correctly (by checking the seed generated by the client against that of the server). One would have been motivated to do so, as it would increase the security of the seed generation protocol by ensuring that neither of the strings generated by the client and servers were intercepted and replaced by an attacker. In doing so, both the client and the server could be assured that the other received the correct string.

As per claim 36, Spies teaches a machine-readable storage medium containing one or more software programs for secure generation of a seed for use in performing one or more cryptographic operations, wherein the one or more software programs when executed by a processing device implement at least one of a seed generation client and seed generation server *(column 3, line 59, the client is a computer; column 4, lines 32-34, the client has an application which implements cryptographic operations, stored in memory and executable on the processor; column 4, lines 23 and 60, both the client and the server have a processor and a memory, which comprises the processing devices as defined within the specification);* wherein the seed generation server provides a SECOND string to the seed generation client *(column 6, lines 53-55, the server generates string, S and sends it to the client; column 8, lines 4-6, the string P and the string S are exchanged between the client and server in an encrypted form)*; the seed generation client generates a FIRST string, encrypts the FIRST string utilizing a key, and sends

the encrypted FIRST string to the seed generation server *(column 2, lines 58-67, the client sends a string P to the server within an encrypted message)*; the seed generation client generates the seed as a function of at least the first string and the second string *(column 7, lines 41-44, four values, including the two strings P and S which were exchanged between the client and server, are combined to serve as the seed),* and the seed generation server decrypts the encrypted FIRST string *(column 3, lines 1-2, the server decrypts the message containing the string P from the client).*

Spies teaches the process of generating a seed initiated by a client. Additionally, Spies teaches the method wherein the seed generation server provides a first string to the seed generation client, as the resulting outcome is the same regardless of whether the client or the server started the seed generation process by providing the first string.

Spies does not explicitly teach apparatus wherein the server independently generates the seed as a function of at least the first string and the second string.

However, it would have been obvious for one of ordinary skill in the art at the time of the invention for the server to independently generate the seed as a function of at least the first string and the second string, in order to validate that the seed was generated by the client correctly (by checking the seed generated by the client against that of the server). One would have been motivated to do so, as it would increase the security of the seed generation protocol by ensuring that neither of the strings generated by the client and servers were intercepted and replaced by an attacker. In doing so, both the client and the server could be assured that the other received the correct string.

As per claim 37, Spies teaches the method for secure generation of a seed for use in performing one or more cryptographic operations, the method being implemented in a seed generation client, the method comprising the steps of: the client generating a first string, encrypting the first string utilizing a key, and sending the encrypted first string to the server *(column 2, lines 58-67, the client sends a string P to the server within an encrypted message)*, receiving a second string from a server *(column 6, lines 53-55, the server generates string, S and sends it to the client; column 8, lines 4-6, the string P and the string S are exchanged between the client and server in an encrypted form)*, and generating the seed as a function of at least the first string and the second string *(column 7, lines 41-44, four values, including the two strings P and S which were exchanged between the client and server, are combined to serve as the seed)*.

Spies teaches the process of generating a seed initiated by a client. Additionally, Spies teaches the process initiated by a server, as the resulting outcome is the same regardless of whether the client or the server started the seed generation process by providing the first string.

As per claim 38, Spies teaches an apparatus for secure generation of a seed for use in performing one or more cryptographic operations, the apparatus comprising: a processing device comprising a processor coupled to a memory, the processing device implementing a seed generation client *(column 4, line 23, the client has a processor and a memory)*; the seed generation client being configured: (i) to receive a SECOND string from a seed generation server *(column 6, lines 53-55, the server generates string, S and sends it to the client; column 8, lines 4-6, the string P and the string S are exchanged between the client and server in an encrypted form)*; (ii) to generate a FIRST string, to encrypt the second string utilizing a key, and to send the encrypted FIRST string to the seed generation server *(column 2, lines 58-67, the client sends a*

string P to the server within an encrypted message); and (iii) to generate the seed as a function of at least the first string and the second string *(column 7, lines 41-44, four values, including the two strings P and S which were exchanged between the client and server, are combined to serve as the seed).*

Spies teaches the process of generating a seed initiated by a client. Additionally, Spies teaches the process initiated by a server, as the resulting outcome is the same regardless of whether the client or the server started the seed generation process by providing the first string.

As per claim 39, Spies teaches a method for secure generation of a seed for use in performing one or more cryptographic operations, the method being implemented in a seed generation server, the method comprising the steps of: providing a SECOND string to a seed generation client *(column 6, lines 53-55, the server generates string, S and sends it to the client; column 8, lines 4-6, the string P and the string S are exchanged between the client and server in an encrypted form)*; receiving from the seed generation client a FIRST string encrypted utilizing a key; decrypting the encrypted FIRST string *(column 2, lines 58-67, the client sends a string P to the server within an encrypted message; column 3, lines 1-2, the server decrypts the message containing the string P)*;

Spies teaches the process of generating a seed initiated by a client. Additionally, Spies teaches the method wherein the seed generation server provides a first string to the seed generation client as the resulting outcome is the same regardless of whether the client or the server started the seed generation process by providing the first string.

Spies does not explicitly teach the server generating the seed as a function of at least the first string and the second string.

However, it would have been obvious for one of ordinary skill in the art at the time of the invention for the server to independently generate the seed as a function of at least the first string and the second string, in order to validate that the seed was generated by the client correctly (by checking the seed generated by the client against that of the server). One would have been motivated to do so, as it would increase the security of the seed generation protocol by ensuring that neither of the strings generated by the client and servers were intercepted and replaced by an attacker. In doing so, both the client and the server could be assured that the other received the correct string.

As per claim 40, An apparatus for secure generation of a seed for use in performing one or more cryptographic operations, the apparatus comprising: a processing device comprising a processor coupled to a memory, the processing device implementing a seed generation server *(column 4, line 60, the server has a processor and a memory)*; the seed generation server being configured: (i) to provide a SECOND string to a seed generation client *(column 6, lines 53-55, the server generates string, S and sends it to the client)*; (ii) to receive from the seed generation client a FIRST string encrypted utilizing a key *(column 2, lines 58-67, the client sends a string P to the server within an encrypted message; column 8, lines 4-6, the string P and the string S are exchanged between the client and server in an encrypted form)*; (iii) to decrypt the encrypted FIRST string *(column 3, lines 1-2, the server decrypts the message containing the string P).*

Spies teaches the process of generating a seed initiated by a client. Additionally, Spies teaches the method wherein the seed generation server provides a first string to the seed generation client, as the resulting outcome is the same regardless of whether the client or the server started the seed generation process by providing the first string.

Spies does not explicitly teach an apparatus implementing a server, wherein the server is configured to generate the seed as a function of at least the first string and the second string.

However, it would have been obvious for one of ordinary skill in the art at the time of the invention for the server to independently generate the seed as a function of at least the first string and the second string, in order to validate that the seed was generated by the client correctly (by checking the seed generated by the client against that of the server). One would have been motivated to do so, as it would increase the security of the seed generation protocol by ensuring that neither of the strings generated by the client and servers were intercepted and replaced by an attacker. In doing so, both the client and the server could be assured that the other received the correct string.

9. Claims 2 and 15 are rejected under 35 U.S.C. 103(a) as being unpatentable over Spies in view of Fielder et al. *(US Patent No. 5963646)* (hereinafter Fielder).

As per claim 2, Spies teaches the method of claim 1 as applied above.

Spies does not teach the method wherein the seed comprises a symmetric key.

However, Fielder teaches the method wherein the seed comprises a symmetric key *(abstract; column 3, lines 19-27, Fielder teaches the method of generating a pseudorandom number that may serve as a deterministic symmetric encryption key).*

It would have been obvious for one of ordinary skill in the art at the time of the invention to combine the teachings of Spies with Fielder in order to generate a seed in a manner similar to a key agreement protocol. In doing so, such a seed would essentially have the same structure as a key (both seeds and keys are just bit strings of varying lengths that are random or pseudorandom). One would have been motivated to do so as Burnett teaches that attackers can

attempt to recover private data throughout the cryptographic process *(pp. 34, attackers can determine a key by going backwards to the components that were used to generate the key, a PRNG, which is generally not well hidden, and the seed).* Therefore, generating a seed in a manner similar to a key agreement protocol ensures that the seed is more resistant to attack.

As per claim 15, Spies teaches the method of claim 1.

Spies does not teach the method wherein the server initiates the seed generation process responsive to receipt of a management command.

However, Fielder teaches the method wherein the seed generation server <u>initiates the seed generation process responsive to receipt of a management command</u> *(column 3, lines 22-33, a seed is generated and applied through a hash function to provide the key; column 7, lines 38-40, an activation code initiates the generation of this process).*

It would have been obvious for one of ordinary skill in the art at the time of the invention to combine the teachings of Spies with Fielder in order to allow the party that submits the management command to direct the generation of the seed as needed. One would have been motivated to do so as this increase level of control allows the seed generation process to be "automated" in a manner that would result in more efficient replacement of seeds on a random basis, therefore providing for the generation of seeds that are more resistant to brute-force attacks.

10. Claims 3, 4, and 29 are rejected under 35 U.S.C. 103(a) as being unpatentable over Spies in view of Burnett et al. *(2001)* (hereinafter Burnett).

As per claims 3 and 4, Spies teaches the method of claim 1 as applied above.

Spies does not teach the method wherein the seed is generated as a function of a combination of the second string and identifying information associated with the seed generation server.

However, Burnett teaches the method wherein the seed is generated as a function of a combination of the second string and identifying information associated with the seed generation server *(pp. 249, Burnett describes the Diffie-Hellman key agreement in which the recipient's public key is combined with the sender's private key to generate a shared key)*. Burnett describes how the public key infrastructure provides a mechanism by which a public key is bound to a user in such a manner as to properly authenticate the user's identity *(pp. 171-172)*. In doing so, the tamperproof binding of the public key to the user allows one to be able to securely rely upon the public key to identify the user who possesses it.

It would have been obvious to one of ordinary skill in the art at the time of the invention that the recipient who contributes a public key in the Diffie-Hellman key agreement may be the server, and that the identifying information associated with the server comprises a public key of the server. Additionally, it would have been obvious to combine the teachings of Spies with that of Burnett in order to provide a manner of authenticating the seed generated by a server through interaction with a client. Doing so would ensure that the generation of the seed is more secure, as the server is guaranteed to have been associated with the generation process.

As per claim 29, Spies teaches the method of claim 1 as applied above. Spies does not explicitly teach the method wherein the generated seed is used to replace an existing seed known to both the seed generation client and the seed generation server.

However, Burnett teaches that pseudo-random number generators are deterministic, such that changing the seed would change the outputted pseudo-random number. Additionally, Burnett teaches that unlike true random number generators, whose input is constantly changing on its own on a random basis, it is important for the user to ensure that the input (the seed) to a pseudo-random number generator to change each time you want to generate a new number to serve as an encryption key *(pp. 28)*. Burnett describes how key management protocols support the function of key updates, through which key pairs used in an asymmetric cryptography system must be updated regularly by replacing key pairs with new ones *(pp. 183)*.

Therefore, it would have been obvious for one of ordinary skill in the art at the time of the invention to combine the teachings of Spies in order to randomly generate new seeds to replace existing seeds, which can then be used as input to key generators to produce new encryption keys. One would have been motivated to do so, as seeds are used as inputs to pseudo-random number generators to produce long strings that may be used as encryption keys, and changing encryption keys on a regular basis provides increased security against brute-force attacks; in the case of encryption keys which are generated from deterministic pseudo-random generators, it is necessary for the input (the seed) to be changed in order to generate new keys which may then replace old keys.

11. Claims 20-21, and 25 are rejected under 35 U.S.C. 103(a) as being unpatentable over Spies in view of Fielder and further in view of Burnett.

As per claim 20, Spies teaches the method of claim 1.

Spies does not teach the method wherein the seed is generated by applying a cryptographic algorithm to an additional string generated utilizing the first string, the second string, and the key.

However, Fielder teaches the method wherein the seed is generated by applying a cryptographic algorithm to an additional string generated utilizing the first string, the second string, and the key *(column 3, lines 50-52, the first string, a constant value, may combined with a second string, the E-Key seed, through a sequence of cryptographic steps to provide an input (seed) to a secure hash function; column 3, lines 53-55, the E-Key seed and constant value may be encrypted).*

It would have been obvious for one or ordinary skill in the art at the time of the invention to combine the teachings of Spies with that of Fielder in order apply a block cipher with a feedback mode by repeatedly apply the cryptographic algorithm to successive portions of the additional string.

One would have been motivated to do so as a block cipher comprises one type of symmetric key algorithm and utilizing a feedback mode solves the problem of copies of ciphertext resulting from applying a block cipher, which an attacker might identify as a repeated pattern *(Burnett, pp. 40).* By repeatedly applying the algorithm to portions of the additional string, the seed appears more random, and therefore becomes more resistant to attacks.

As per claim 21, Spies, Fielder, and Burnett teach the method of claim 20 as applied above. Additionally, Fielder further teaches the method wherein the additional string generated utilizing the first string, the second string and the key comprises a concatenation of the first string, the second string, and the key *(column 3, lines 49-52, a constant value, the first string,*

may be combined with the E-Key seed, the second string, through a sequence of logic, algebraic, and/or cryptographic steps). It would have been obvious to one of ordinary skill in the art at the time of the invention to concatenate the first string, the second string, and the key prior to applying a cryptographic algorithm to the generated string in order to produce a seed as concatenation is one of the simplest methods of combining two bit sequences.

As per claim 25, Spies, Fielder, and Burnett teach the method of claim 20 as applied above. Additionally, Fielder teaches the method wherein the <u>cryptographic algorithm comprises an encryption operation</u> *(column 2, lines 23-25, encryption algorithms are required to generate an encryption key, which may be used as a seed, as stated earlier)*.

12. Claims 16-17 are rejected under 35 U.S.C. 103(a) as being unpatentable over Spies in view of Saito et al. *(US Patent No. 6125186)* (hereinafter Saito).

As per claim 16, Spies teaches the method of claim 1, but does not teach the method wherein <u>the seed generation server initiates the seed generation process responsive to receipt of a request initiated by the seed generation client.</u>

However, Saito teaches a server establishing an encrypted communication path after receiving a message from the client *(Figure 4A, client sends notice of start-up completion)*, which is followed by the actual encryption and decryption processing to be performed.

It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Spies with Saito, as it is necessary to establish an encrypted communication path before any processes may be initiated. For cryptographic protocols, it is important to change the key needed for encryption in order to increase security *(column 3, lines 28-30)*. In addition, for protocols utilizing deterministic keys which can be independently

generated, changing a key produced by a pseudo-random number generator necessitates changing the seed used for input. Therefore, optimally secure cryptographic protocols would generate a new session key (and therefore a new seed) upon the establishment of a new connection between the server and client.

As per claim 17, Spies and Saito teach the method of claim 16 as applied above. Saito further teaches the method of generating a seed in which the seed generation client provides the seed generation server with information indicating one or more processing algorithms suitable for use in the seed generation process *(column 6, lines 18-21; an instruction signal is sent which indicates which cryptographic processing unit corresponding to a particular algorithm is to be used)*.

13. Claims 7-8, 11-12, 14, and 28 are rejected under 35 U.S.C. 103(a) as being unpatentable over Spies in view of Kaliski, Jr. *(US Pre-Grant Publication 2001/0055388)* (hereinafter Kaliski).

As per claim 7, Spies teaches the method of claim 1, but does not teach the method wherein the seed generation client comprises or is otherwise associated with an authentication token.

However, Kaliski teaches *(paragraph [0006-0007], private data may be stored on a token that is physically connected to a client)* a client which is connected to an authentication token.

It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Spies with Kaliski in order to provide a device which may hold private data which combined with other data may generate a seed. One would have been

motivated to do so as utilizing a token allows for a more secure generation protocol, resulting in an attacker having less opportunity to attempt to steal the private data used by the client, as data on a token is only accessible when connected to a client. Should an attacker compromise a client, they may not necessarily gain access to the private data since it is not stored on the client itself, but rather on a token.

As per claim 8, Spies teaches the method of claim 1, but does not teach the method <u>wherein the seed generation server comprises or is otherwise associated with an authentication entity.</u>

However, Kaliski teaches *(paragraph [0019], Kaliski teaches the use of verification servers, which may or may not also be the servers together with a client generate a strong secret, which may be used as a seed)* a server which comprises or is otherwise associated with an authentication entity. Kaliski describes verification servers which provide authentication of the regenerated strong secret.

It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Spies with that of Kaliski in order to provide a mechanism for authentication of a generated seed created by deterministic means *(paragraph [0019], Kaliski describes how authentication could help determine if an unauthorized entity is attempting to regenerate the strong secret)*. One would have been motivated to do so as an authenticated seed provides for a more secure seed generation and consequently key generation.

As per claim 11, Spies teaches the method of claim 1, but does not teach the method <u>wherein the seed generation server sends the generated seed to an authentication entity.</u>

However, Kaliski teaches (*paragraph [0019], Kaliski teaches the use of verification servers, which may or may not also be the servers together with a client generate a strong secret, which may be used as a seed)* a server which comprises or is otherwise associated with an authentication entity. Kaliski describes verification servers which provide authentication of the regenerated strong secret.

It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Spies with that of Kaliski in order to provide a mechanism for authentication of a generated seed created by deterministic means *(paragraph [0019], Kaliski describes how authentication could help determine if an unauthorized entity is attempting to regenerate the strong secret).* One would have been motivated to do so as an authenticated seed provides for a more secure seed generation and consequently key generation. In addition, in the case where the authentication entity may not be the same as the seed generation server, it is clear that there needs to be a way for the server to send the generated seed to the authentication entity to perform appropriate authentication.

As per claim 12, Spies and Kaliski teach the method of claim 11 as applied above. Kaliski further teaches the method <u>wherein the seed generation server also sends user identifying information associated with the seed to the authentication entity.</u> It would have been obvious to one of ordinary skill in the art at the time of the invention to send user identifying information associated with the data to be authenticated. In doing so, one would be able to determine if the seed was regenerated by the appropriate entities by checking the regenerated seed against a database or other entity that has bound the seed with an identity, as is done in public key infrastructure.

As per claim 14, Spies teaches the method of claim 1, but does not teach the method wherein the seed generation client and the seed generation server communicate with one another through at least one intermediary processing device.

However, Kaliski teaches the method wherein the client in concert with an intermediary processing device provides a string to be used to generate a strong secret *(paragraph [0054], a generating client takes data stored in a token, the intermediary processing device, in order to generate a weak secret, which is then combined with a server's string in order to produce a strong secret)*. It is clear that a token may communicate with a server via an intermediary processing device, such as a client, which is able to read the data stored on a token.

It would have been obvious to one of ordinary skill in that art at the time of the invention to combine the teaching of Spies with Kaliski in order to provide for a more secure generation protocol, as an attacker has less opportunity to attempt to steal the private data used by the client, as data on a token is only accessible when connected to a client. Should an attacker compromise a client, they may not necessarily gain access to the private data since it is not stored on the client itself, but rather on a token.

As per claim 28, Spies teaches the method of claim 1, but does not teach the method wherein the seed generation server stores the generated seed in an authentication entity.

However, Kaliski teaches *(paragraph [0019], Kaliski teaches the use of verification servers, which may or may not also be the servers together with a client generate a strong secret, which may be used as a seed)* a server which comprises or is otherwise associated with an authentication entity. Kaliski describes verification servers which provide authentication of the regenerated strong secret.

It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Spies with that of Kaliski in order to provide a mechanism for authentication of a generated seed created by deterministic means *(paragraph [0019], Kaliski describes how authentication could help determine if an unauthorized entity is attempting to regenerate the strong secret)*. One would have been motivated to do so as an authenticated seed provides for a more secure seed generation and consequently key generation.

In addition, in the case where the authentication entity may not be the same as the seed generation server, it is clear that there needs to be a way for the server to send the generated seed to the authentication entity to perform appropriate authentication. Kaliski teaches that the server may store the generated seed in an authentication entity *(paragraph [0016], Kaliski describes how the verification server[s] may store verifier data)*. Once the generated seed is sent to the authentication entity, the entity will be able to determine if the identity may be verified by referring to some record previously stored in the authentication entity that has evidence of the seed bound to a user, similar to the authentication of public keys in PKI.

14. Claims 9 and 27 are rejected under 35 U.S.C. 103(a) as being unpatentable over Spies in view of Yatsukawa *(US Patent No. 61480404)*.

As per claim 9, Spies teaches the method of claim 1, but does not teach the method <u>wherein the seed generation server sends an authentication code to the seed generation client, the authentication code proving knowledge of the generated seed and instructing the seed generation client to store the generated seed.</u>

However, Yatsukawa teaches the method wherein the client stores the generated seed upon receipt of an authentication code by the server *(Figure 13, the client stores authentication*

data D_2 upon receiving a message of "grant" indicating the authentication processing result from the server). Notification of grant of the authentication request received from the authentication server assures that both the server's knowledge of the generated authentication data matches that of the client *(column 13, lines 23-29).*

It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the teaching of Spies with Yatsukawa in order to ensure that the appropriate seed is stored by the client. Such an authentication method would make it difficult for an unauthorized entity to replace the seed which was securely generated with a false seed right before it is stored.

As per claim 27, Spies teaches the method of claim 1, but does not teach the method wherein the seed generation client stores the generated seed in an authentication token.

However, Yatsukawa teaches the client storing a seed *(Figure 13, the client stores seed data used for generating authentication data).* Yatsukawa also teaches a client which is associated with an authentication token *(Figure 5).*

It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teaching of Spies with Yatsukawa in order to provide increased security in generating a seed, as storing personal data such as the seed within a token allows it to be managed relatively safely and makes "masquerading" by an unauthorized entity in the generation protocol generally difficult *(Yatsukawa, column 9, lines 45-51).*

15. Claims 23-24 are rejected under 35 U.S.C. 103(a) as being unpatentable over Spies in view of Fielder and further in view of Burnett, and further in view of Carro et al. *(US Pre-Grant Publication 2002/0013794)* (hereinafter Carro).

As per claims 23 and 24, Spies, Fielder, and Burnett teach the method of claim 20, but do not teach the method wherein the cryptographic algorithm comprises a one- way cryptographic operation.

However, Carro teaches the generating a seed *(Figure 3, a seed is generated from applying a function to a string 'cText' with a secret key)* from two strings (a secret-key and a string 'cText') through the use of one-way hashing *(paragraph [0026])*.

It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teachings of Spies, Fielder, and Burnett with Carro, as one-way hashing provides a mechanism in which a deterministic value may be generated such that the same seed is not produced from different combinations of strings.

16. Claim 10 is rejected under 35 U.S.C. 103(a) as being unpatentable over Spies in view of Yatsukawa, and further in view of Carro.

As per claim 10, Spies and Yatsukawa teach the method of claim 9 as applied above, but do not teach the method wherein the authentication code is cryptographically derived from a secret key shared by the seed generation client and the seed generation server.

More specifically, Yatsukawa teaches enciphering seed data by a secret key *(column 11, lines 40-43)* in order to generate an authentication code sent from one party to another in order to provide authentication. The authentication code taught by Yatsukawa was derived from a private key of an asymmetric key pair.

However, Carro teaches that one type of authentication code, known as a MAC, is often computed from a secret key shared only by the sender and receiver *(paragraph [0003])*. It would have been obvious for one of ordinary skill in the art at the time of the invention to modify the

teachings of Spies and Yatsukawa with Carro in order to cryptographically derive the authentication code from a secret key, rather than a private key associated with the client.

17. Claims 30-34 are rejected under 35 U.S.C. 103(a) as being unpatentable over Spies in view of Burnett, and further in view of Yatsukawa and further in view of Kaliski.

As per claim 30, Spies and Burnett teach the method of claim 29, but do not teach the method <u>wherein the generated seed is used to replace an existing seed in an authentication token associated with the seed generation client and in an authentication entity associated with the seed generation server.</u>

However, Yatsukawa teaches the client storing a seed *(Figure 13, the client stores seed data used for generating authentication data)*. Yatsukawa also teaches a client which is associated with an authentication token *(Figure 5)*.

It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teachings of Spies and Burnett with Yatsukawa in order to provide increased security in generating a seed, as storing personal data such as the seed within a token allows it to be managed relatively safely and makes "masquerading" by an unauthorized entity in the generation protocol generally difficult *(Yatsukawa, column 9, lines 45-51)*.

Kaliski teaches that the server may store the generated seed in an authentication entity *(paragraph [0016], Kaliski describes how the verification server[s], which provide authentication of the regenerated strong secret, may store verifier data)*. Once the generated seed is sent to the authentication entity, the entity will be able to determine if the identity may be verified by referring to some record previously stored in the authentication entity that has evidence of the seed bound to a user, similar to the authentication of public keys in PKI.

It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teachings of Spies, Fielder, Burnett, and Yatsukawa with that of Kaliski in order to provide a mechanism for authentication of a generated seed created by deterministic means *(paragraph [0019], Kaliski describes how authentication could help determine if an unauthorized entity is attempting to regenerate the strong secret)*. Additionally, it would have been obvious replace an existing seed in an authentication token associated with the seed generation client and in an authentication entity associated with the seed generation server, for increased security, as stated earlier.

As per claim 31, Spies, Burnett, Yatsukawa, and Kaliski teach the method of claim 30 as applied above. Yatsukawa further teaches the method wherein the authentication token replaces the existing seed with the generated seed after the receipt of a signal from the authentication entity *(Abstract, upon receiving a grant from the server, which performs authentication, the client stores the data as seed data in place of the first seed data)*.

As per claim 32, Spies, Burnett, Yatsukawa, and Kaliski teach the method of claim 31 as applied above. Yatsukawa further teaches the method wherein the signal comprises an authentication code cryptographically derived from the seed *(column 11, lines 40-43, Yatsukawa teaches enciphering seed data by a secret key in order to generate authentication data sent from one party to another in order to provide authentication; Figure 13, after comparison of the authentication data, the client/server stores the new seed data in the place of the old one)*.

As per claim 33, Spies, Burnett, Yatsukawa, and Kaliski teach the method of claim 30 as applied above. Yatsukawa further teaches the method wherein the existing seed is replaced with the generated seed after receipt of a signal. *(Abstract, upon receiving a grant from the server,*

which performs authentication, the client stores the data as seed data in place of the first seed data). Yatsukawa teaches replaces the existing seed in a client upon authentication from a server, but does not teach an authentication entity replacing the seed after receipt of a signal from the authentication token.

However, it would have been obvious to one of ordinary skill in the art at the time of the invention to utilize two-way authentication, as Kaliski teaches that it is important for source of each message exchanged between server and client to be authenticated in order to maintain the security of the overall protocol *(paragraph [0055]).* Therefore, once the token, or the client associated with the token, authenticates the source of the message from the server, it may then instruct the authentication entity associated with the server to replace the seed.

As per claim 34, Spies, Burnett, Yatsukawa, and Kaliski teach the method of claim 33 as applied above. Yatsukawa further teaches the method wherein the signal comprises an authentication code cryptographically derived from the seed *(column 11, lines 40-43, Yatsukawa teaches enciphering seed data by a secret key in order to generate authentication data sent from one party to another in order to provide authentication; Figure 13, after comparison of the authentication data, the client/server stores the new seed data in the place of the old one).*

18. Claims 22 and 26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Spies in view of Fielder and further in view of Burnett, and further in view of Scheidt et al. *(US Pre-Grant Publication 2002/0062451)* (hereinafter Scheidt).

As per claim 22, Spies, Fielder, and Burnett teach the method of claim 20 as applied above, but do not specifically teach the method wherein the additional string comprises n portions C[1], C[2],... C[n], and the seed is generated by computing:

I[1]-- Algorithm (C[1], C[2])

I[2] = Algorithm (I[1], C[3])

...

I[n-1] = Algorithm (I[n-2], C[n])

seed = I[n- 1],

where Algorithm (A, B) denotes application of the cryptographic algorithm to portion B of the string utilizing an algorithm parameter denoted by A.

However, Scheidt teaches the method wherein a working key is constructed from several pieces of information via a combiner function *(paragraph [0056])*. This working key functions similarly to a seed, in that it is used to initialize a symmetric key cryptographic algorithm whereas a seed is used more specifically to initialize a pseudo-random number generator (which comprises of some type of cryptographic algorithm). In addition, the structure of the working key as taught by Scheidt is substantially similar. As such, the method to generate a working key could also be used to generate a seed. Scheidt teaches the working key (which may be used as a seed) generated by applying a combiner function such as Triple DES in CBC Mode *(Figure 5)*. CBC Mode is a type of feedback mode. The algorithm claimed in 22 demonstrates a type of block cipher utilizing a type of feedback mode. It would have been obvious for one of ordinary skill in the art at the time of the invention that rather than using an IV as an algorithm parameter, the algorithm could be applied to the second portion of the string, with the first string functioning as the IV instead. Utilizing the first string as the first parameter eliminates the need to generate a separate value to be used as the IV.

Additionally, it would have been obvious for one of ordinary skill in the art at the time of the invention to modify the teachings of Spies, Fielder, and Burnett with that of Scheidt, as utilizing "splits," or components, in the manner taught by Scheidt to generate a working key, may be used similarly with seed generation, in order to increase the security of the process, as the seed will not be compromised if one of the entities is compromised. This is important as attackers may discover keys produced using a pseudorandom number generator if the seed is compromised *(Burnett, pp. 34)*.

As per claim 26, Spies, Fielder, and Burnett teach the method of claim 25 as applied above, but do not specifically teach the method wherein <u>the encryption operation comprises the AES algorithm.</u>

However, Scheidt teaches the method wherein a working key is constructed from several pieces of information via a combiner function *(paragraph [0056])*. This working key functions similarly to a seed, in that it is used to initialize a symmetric key cryptographic algorithm whereas a seed is used more specifically to initialize a pseudo-random number generator (which comprises of some type of cryptographic algorithm). In addition, the structure of the working key as taught by Scheidt is substantially similar. As such, the method to generate a working key could also be used to generate a seed. Scheidt teaches the working key (which may be used as a seed) generated by applying a combiner function such as Triple DES in CBC Mode *(Figure 5)*.

It would have been obvious for one of ordinary skill in the art at the time of the invention to generate a seed by repeatedly applying the AES algorithm to an additional string generated utilizing the first string, the second string, and the key, as AES is simply the new standard which was created to replace Triple DES. One would apply AES rather than Triple DES, as AES is

more resistant to brute-force attacks and is not as slow. In addition, Scheidt teaches that any other symmetric key algorithm could be substituted for the Triple DES algorithm *(paragraph [0070])*.

It would have been obvious for one of ordinary skill in the art at the time of the invention to modify the teachings of Spies, Fielder, and Burnett with that of Scheidt, as utilizing "splits," or components, in the manner taught by Scheidt to generate a working key, may be used similarly with seed generation, in order to increase the security of the process, as the seed will not be compromised if one of the entities is compromised. This is important as attackers may discover keys produced using a pseudorandom number generator if the seed is compromised *(Burnett, pp. 34)*.

19. Claim 18 is rejected under 35 U.S.C. 103(a) as being unpatentable over Spies in view of Saito, and further in view of Schmeh (2003).

As per claim 18, Spies and Saito teach the method of claim 17 as applied above, but do not explicitly teach the method wherein the seed generation server responsive to the information indicating one or more processing algorithms provides to the seed generation client additional information specifying one or more characteristics of the seed generation process.

However, Schmeh teaches that one common property of a protocol is that of negotiation ability, in which two parties may agree on certain parameters *(pp. 168)*. It would have been obvious for one of ordinary skill in the art at the time of the invention to combine the teachings of Spies and Saito with Schmeh in order for the client and server to exchange information that may be necessary to use a certain algorithm agreed upon in the seed generation process. One would have been motivated to do so as utilizing negotiations in a protocol to specify

characteristics of the seed generation process would allow for increased flexibility for performing the protocol *(Schmeh, pp. 168)*.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to VIRGINIA HO whose telephone number is (571)270-7309. The examiner can normally be reached on Mon to Thu; 7:30 AM - 5:00 PM (Eastern).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Thomas Pham can be reached on 571-272-3689. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/VIRGINIA HO/
Examiner, Art Unit 4148

V.H.

/THOMAS PHAM/
Supervisory Patent Examiner, Art Unit 4148